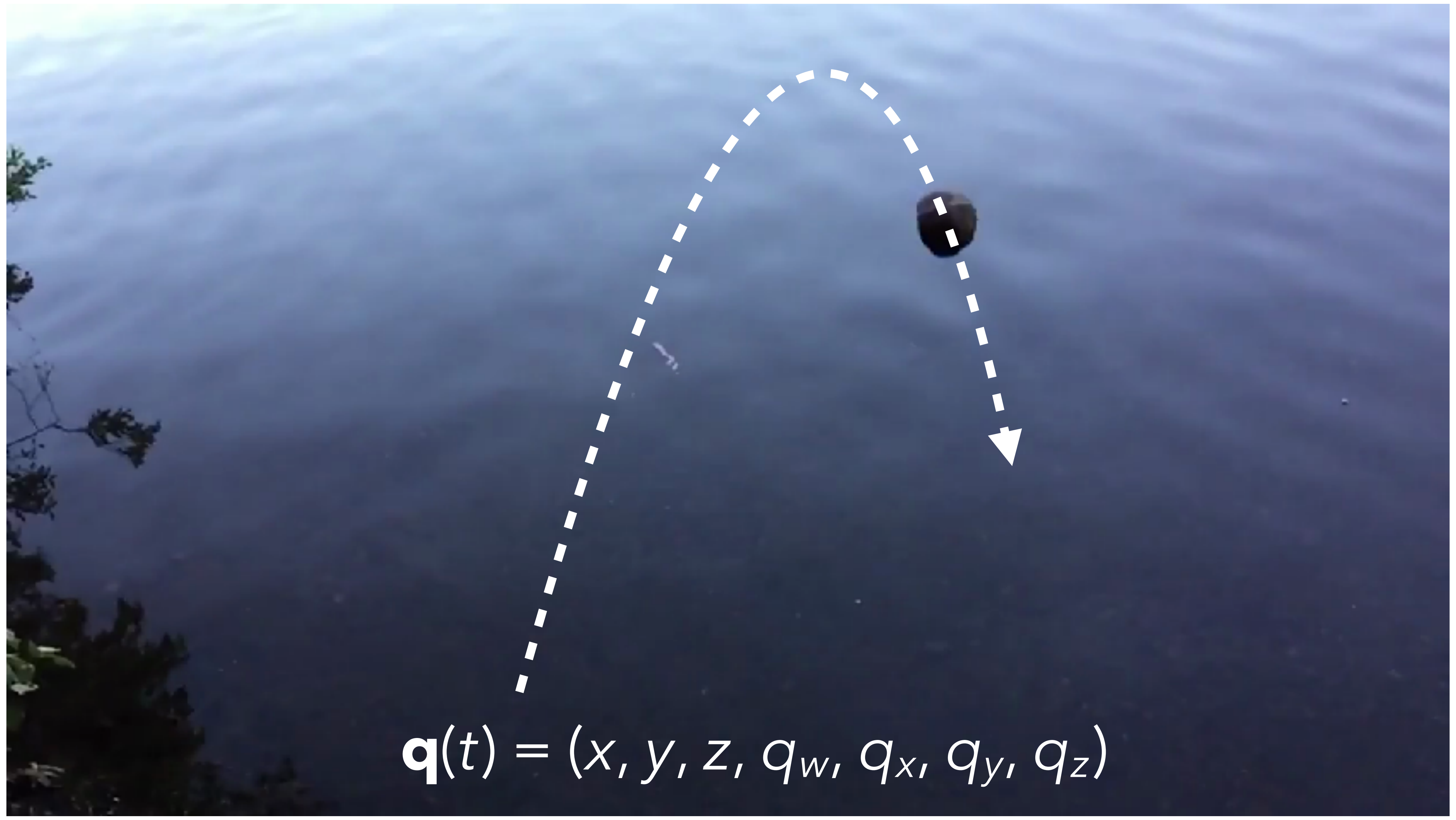
COL781: Computer Graphics

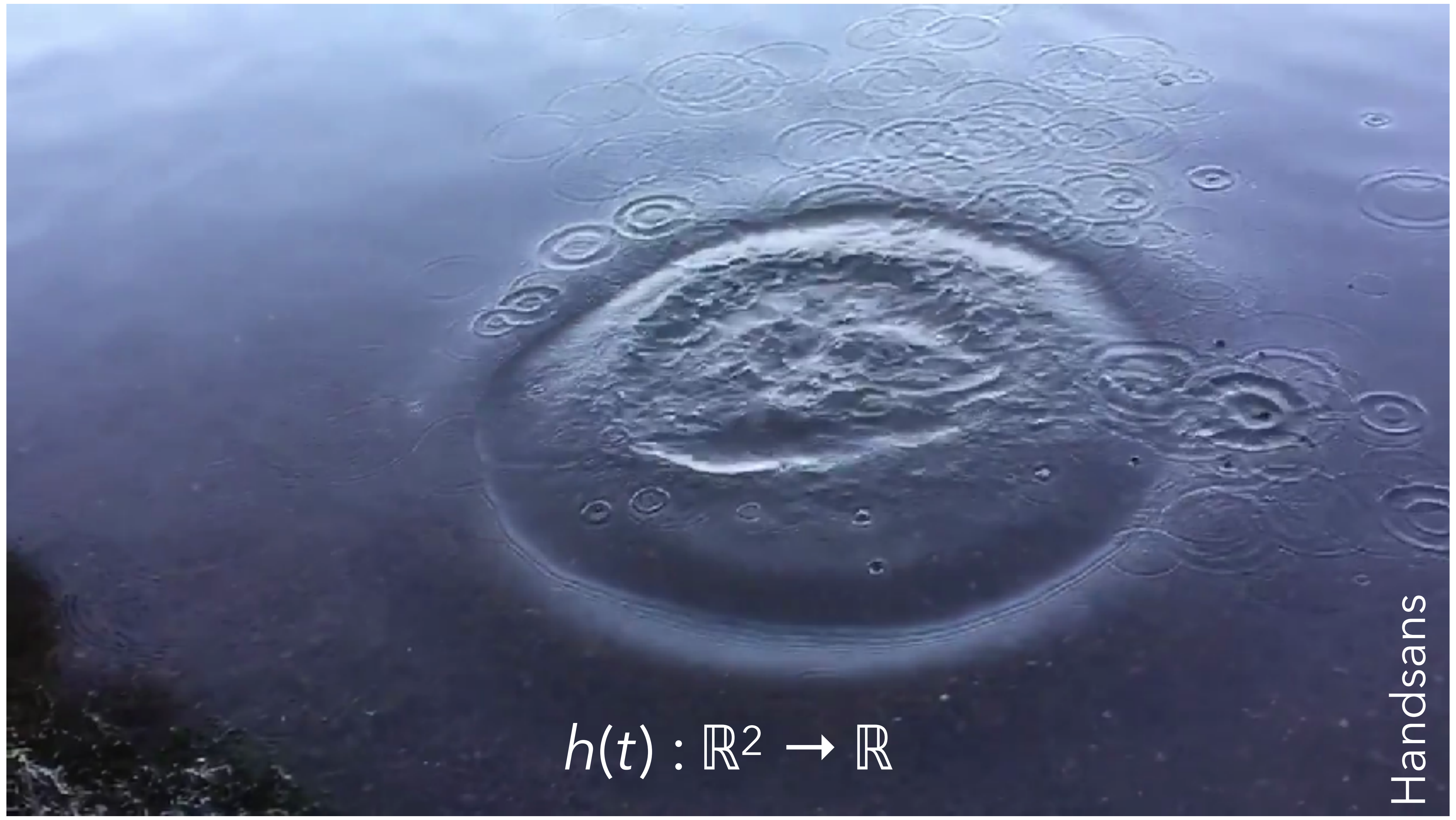
36. Continuum Models

Schreck et al. 2019

So far, we know how to simulate discrete systems of particles and rigid bodies.

But lots of things in real life are not discrete:





$$z = h(t, x, y)$$

Equation of motion will be partial differential equations: temporal derivatives are given in terms of spatial derivatives, e.g.

$$\frac{\partial^2 h}{\partial t^2} = c\left(\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2}\right)$$

Liquids

https://www.youtube.com/watch?v=jVxYuPeqOPI

Narita & Ando 2022

Smoke

https://research.nvidia.com/publication/2008-07_low-viscosity-flow-simulations-animation

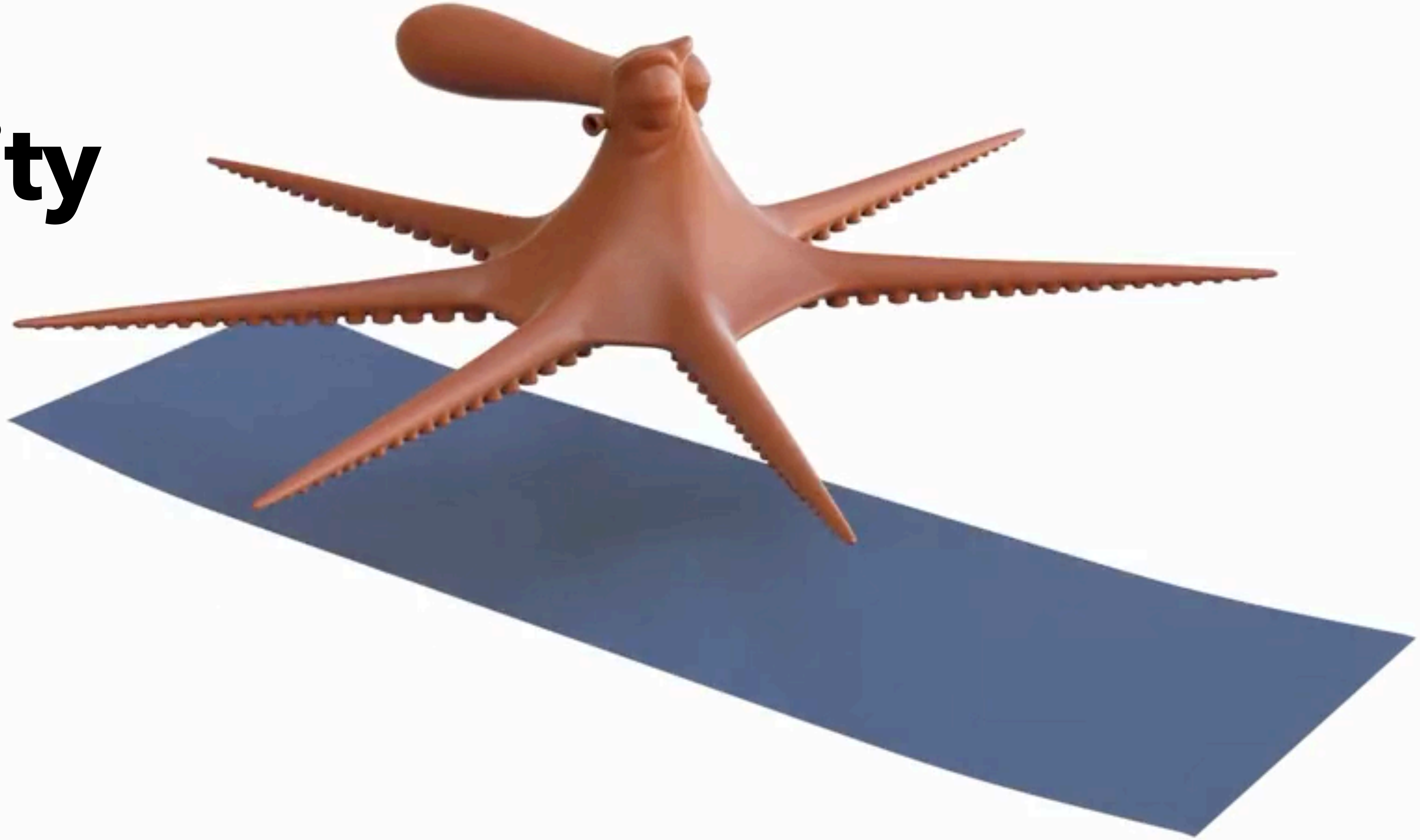
Molemaker et al. 2008

Cloth

https://www.youtube.com/watch?v=TH5g8TuKIkk

Wu et al. 2022

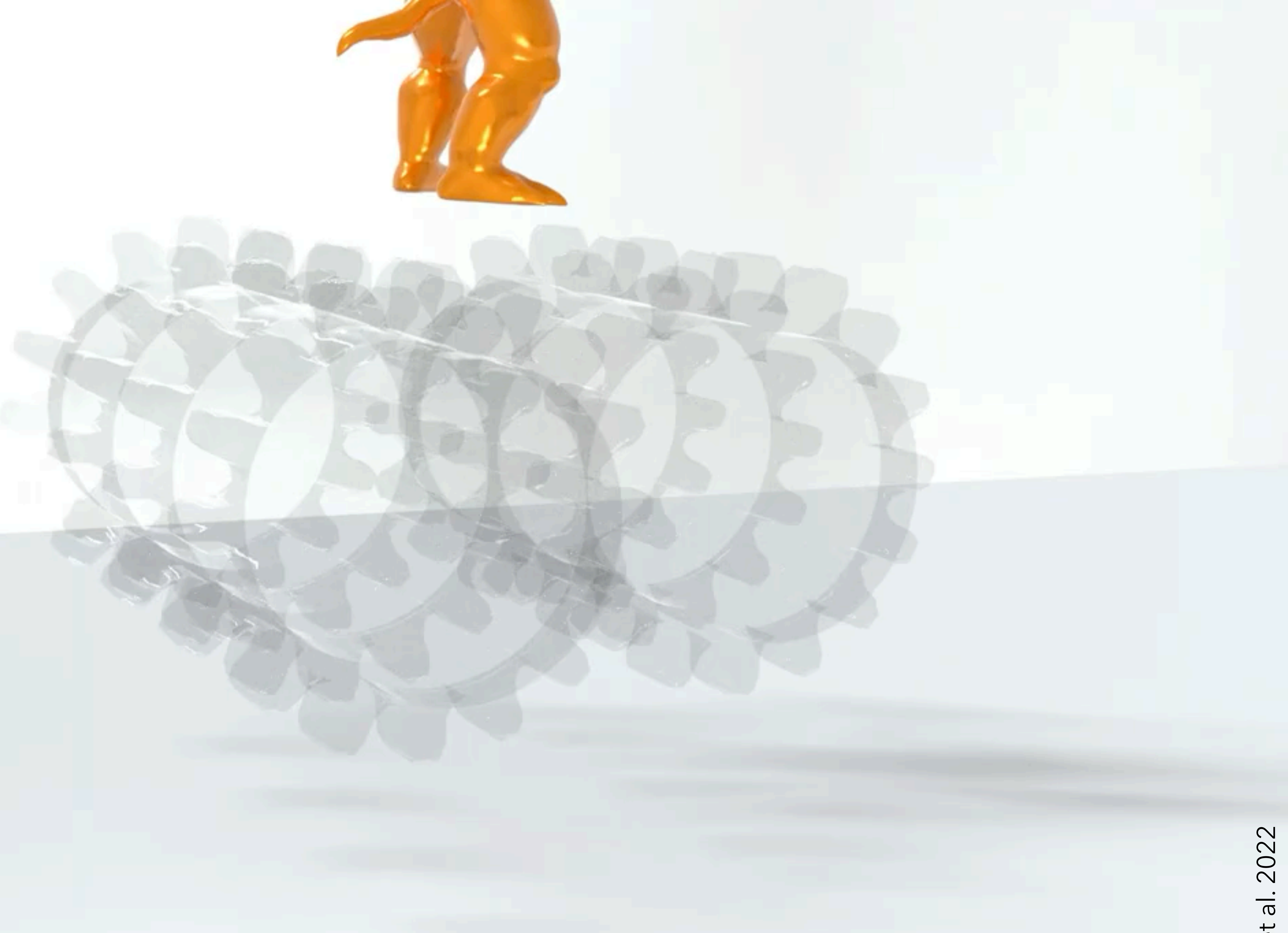
Elasticity



https://vimeo.com/333798247

Kim et al. 2019

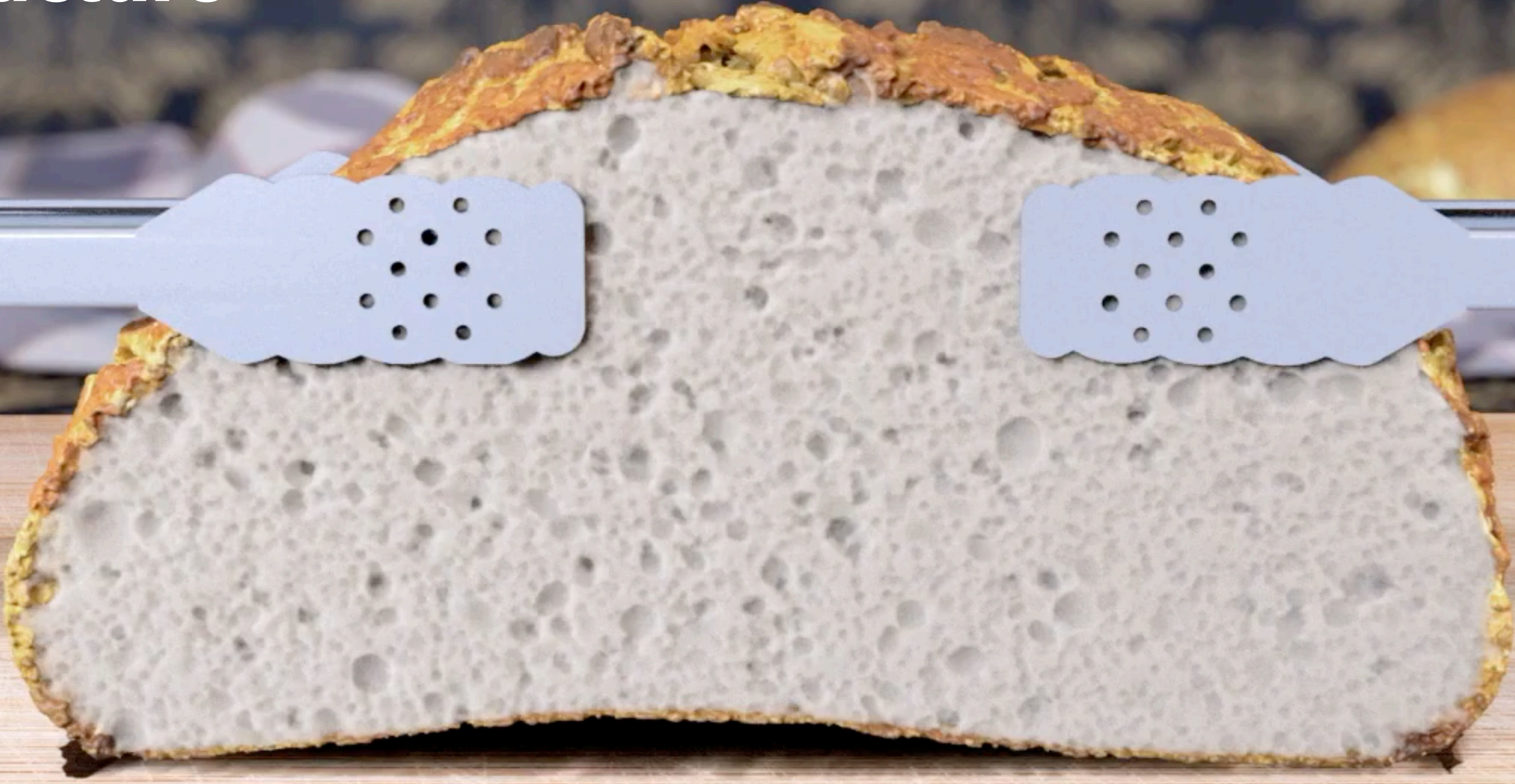
Plasticity



https://www.youtube.com/watch?v=YvvoSu8NK3A

Li et al. 2022

Fracture



https://www.youtube.com/watch?v=lNri-x2nK7o

Wolper et al. 2019

Snow

https://vimeo.com/160322962

Stomakhin et al. 2013

Partial differential equations

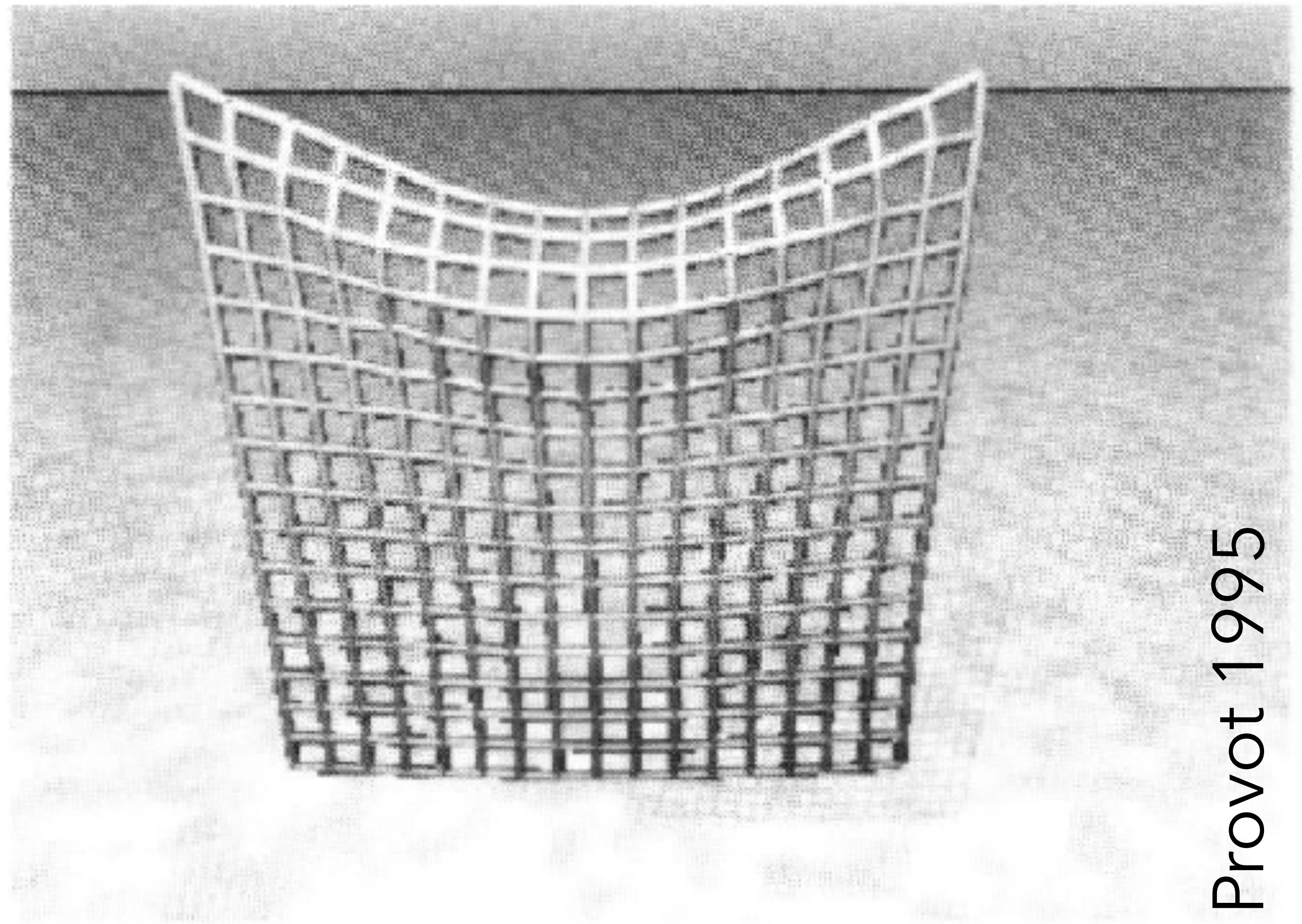
In a mass-spring system, acceleration of particle depends on state of adjacent particles:

$$\ddot{\mathbf{x}}_i = \mathbf{f}(\ldots, \mathbf{x}_{i,j} - \mathbf{x}_{i-1,j}, \ldots, \mathbf{x}_{i,j+1} - \mathbf{x}_{i,j}, \ldots)$$

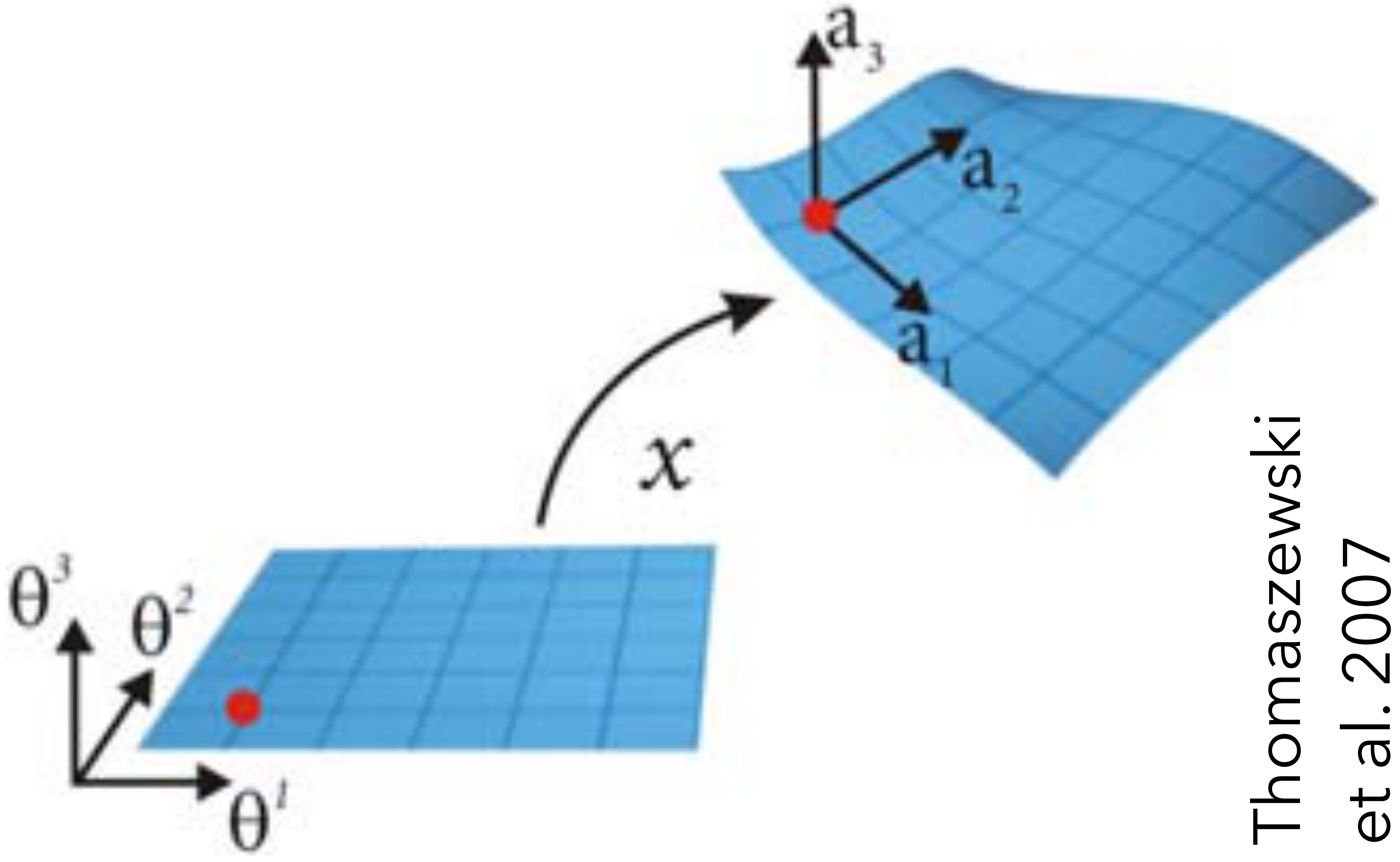
Instead of discrete particles, what if we have a continuous function $\mathbf{x}(u, v)$?

$$\mathbf{x}_{i,j} - \mathbf{x}_{i-1,j}, \ldots \rightarrow \frac{\partial \mathbf{x}}{\partial u}, \frac{\partial \mathbf{x}}{\partial v}$$

$$\frac{\partial^2 \mathbf{x}}{\partial t^2} = \mathbf{f}\left(\ldots, \frac{\partial \mathbf{x}}{\partial u}, \ldots, \frac{\partial \mathbf{x}}{\partial v}, \ldots\right)$$



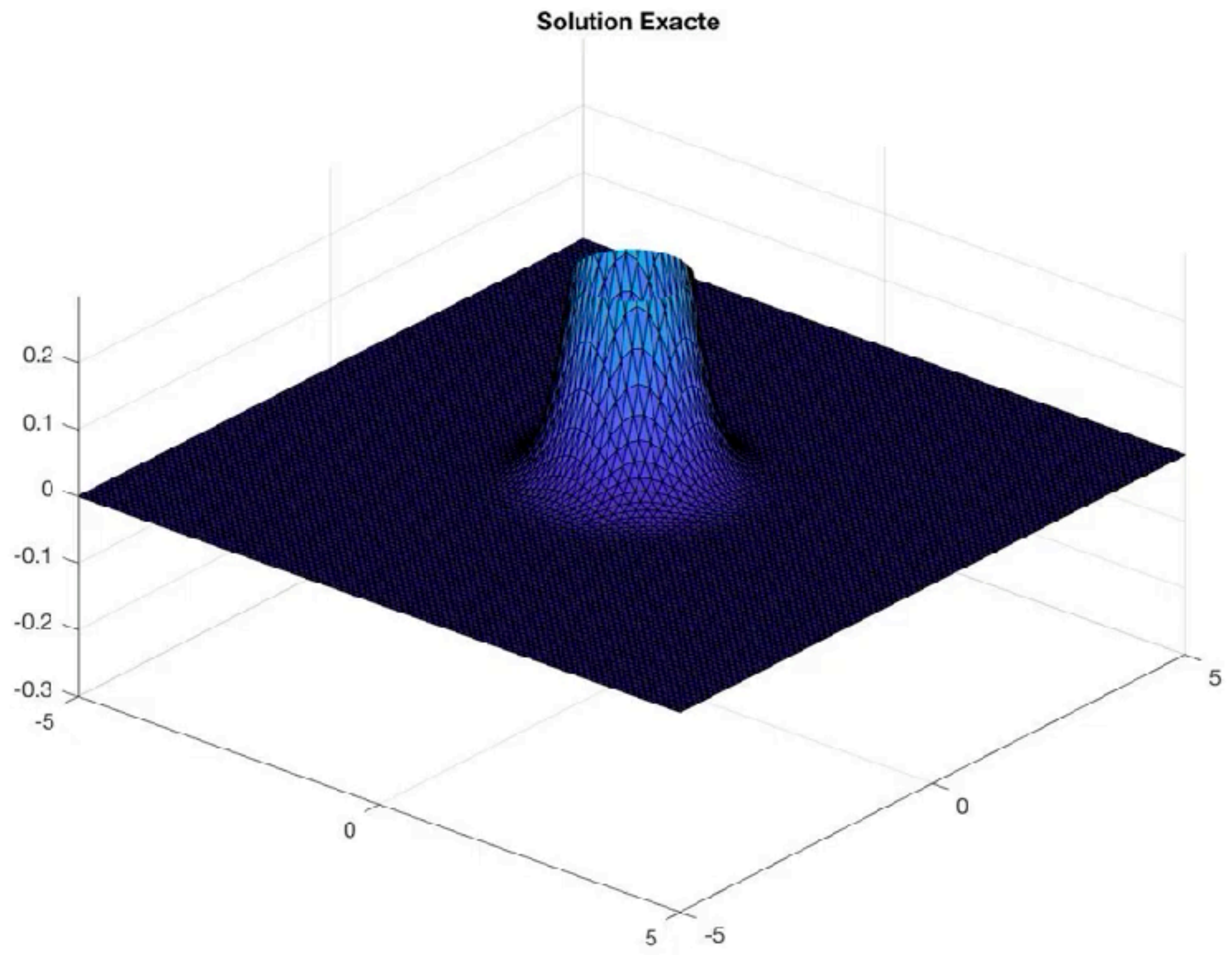
Provot 1995



Thomaszewski et al. 2007

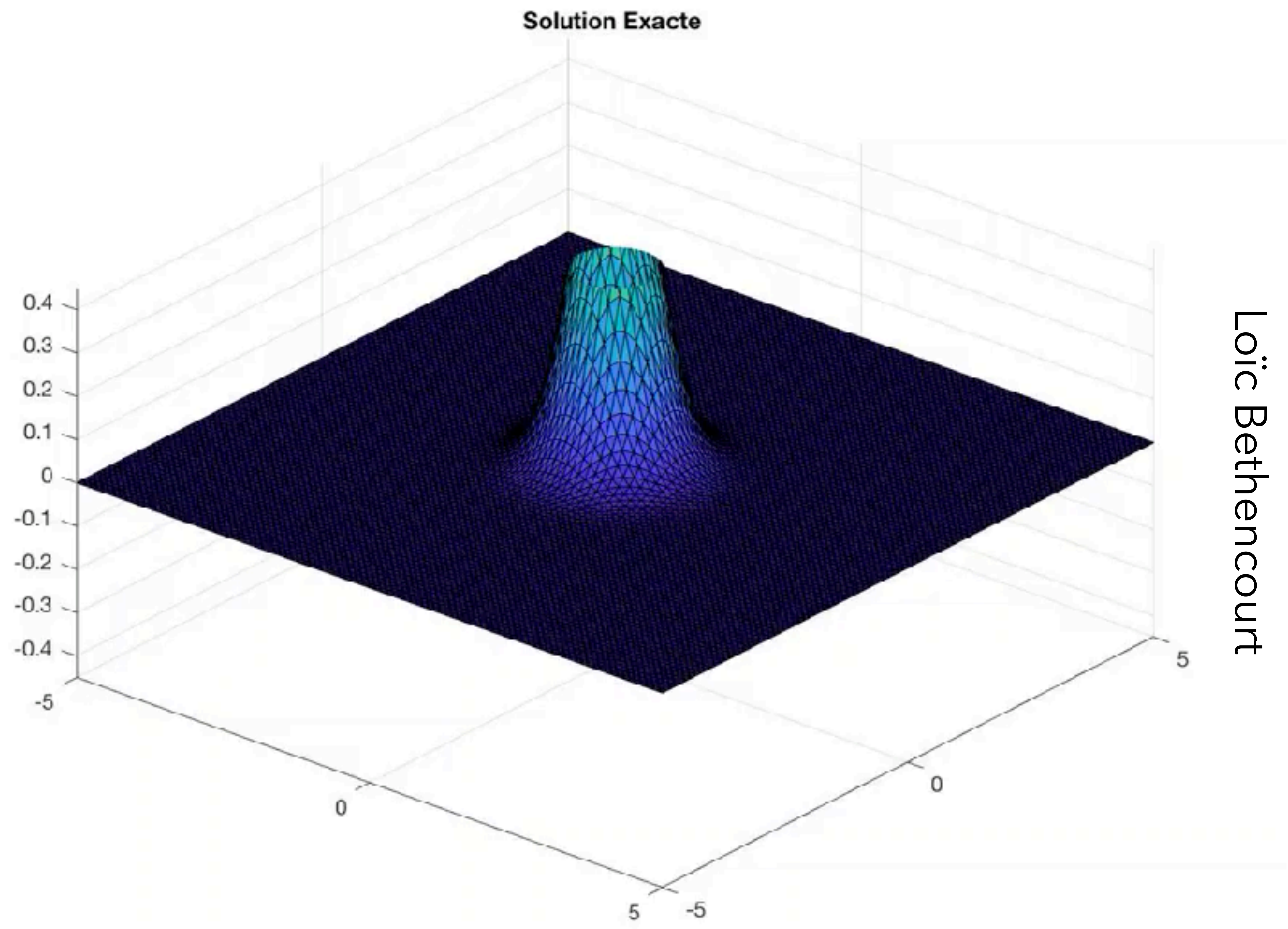
We also need to specify **boundary conditions**:
What happens at the ends of the domain where we can't evaluate $\partial u/\partial x$ etc.?

Dirichlet BC: u = specified



https://www.youtube.com/watch?v=-chMgHvZxH0

Neumann BC: $(\nabla u)_\perp$ = specified



https://www.youtube.com/watch?v=1hsj10dOgt0

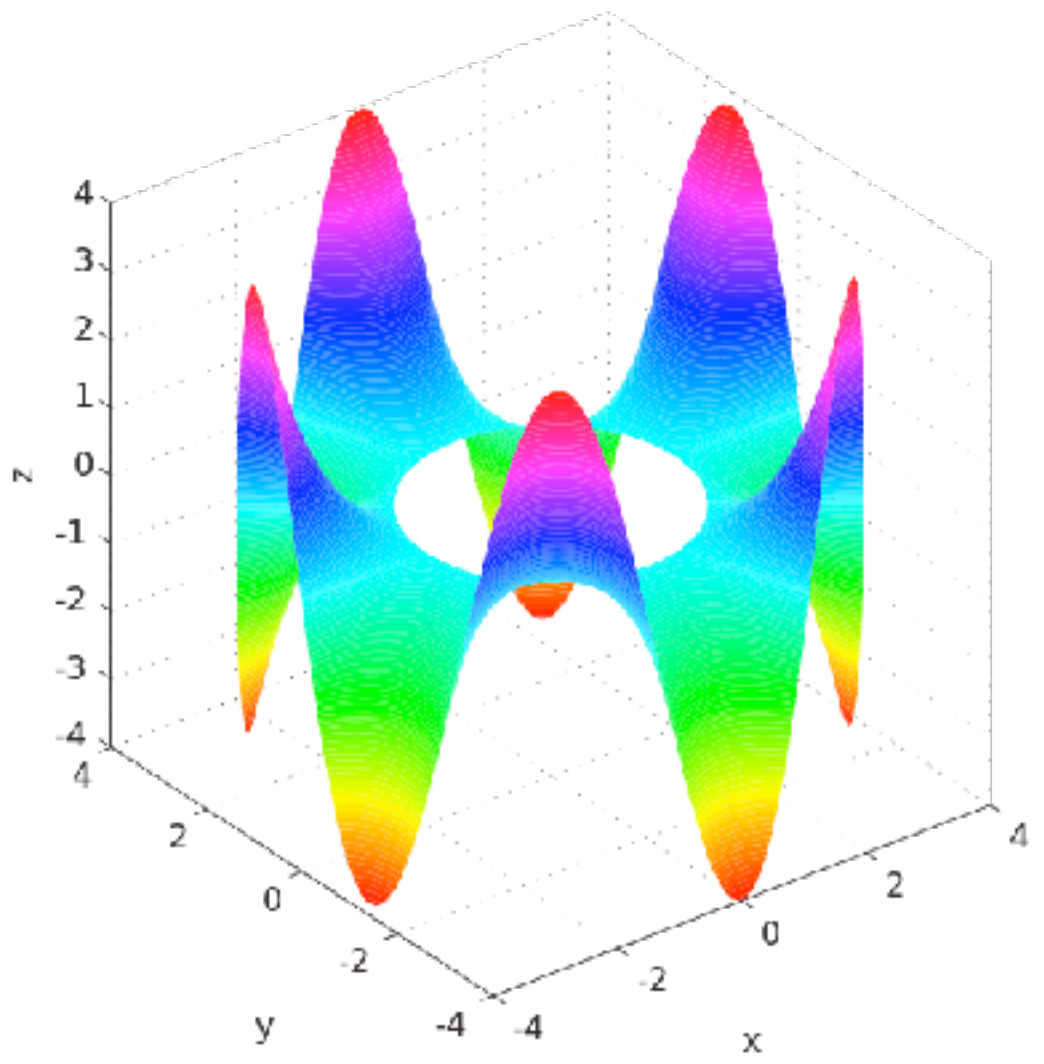
Loïc Bethencourt

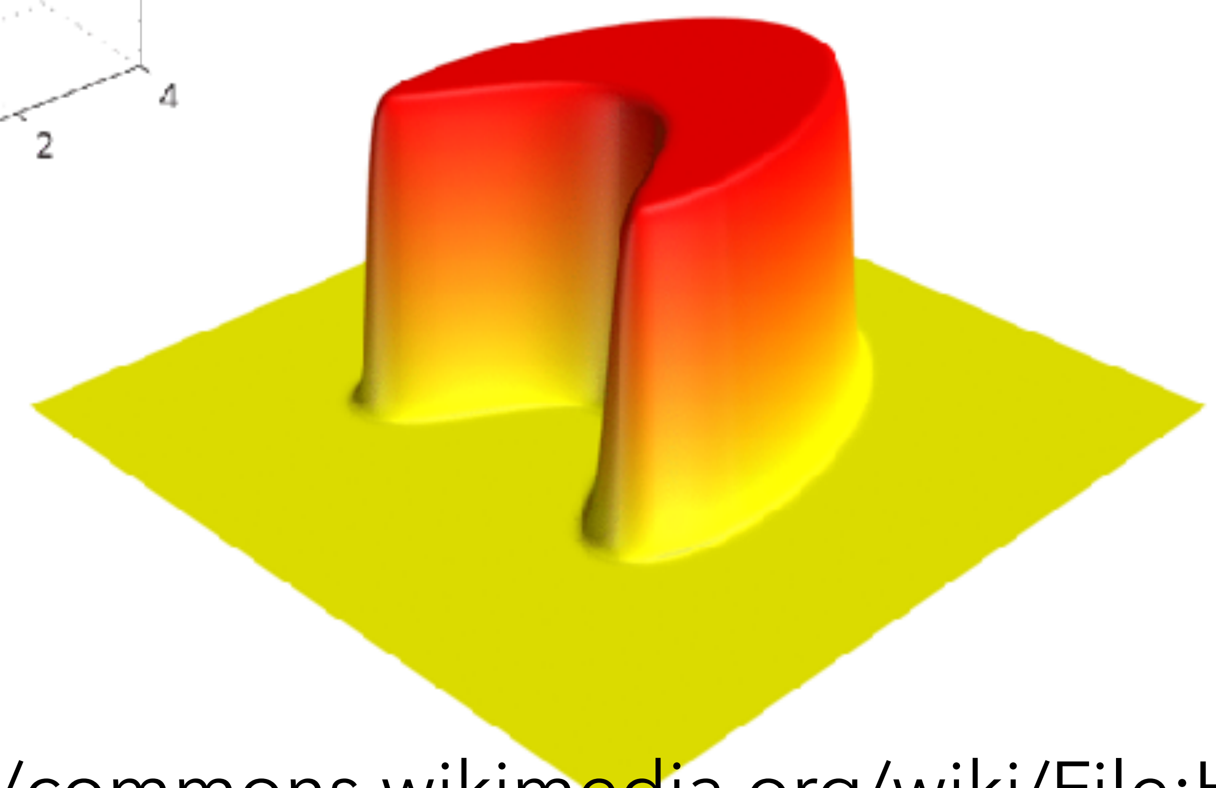
Let's understand a few prototypical PDEs:

- Laplace equation: $\nabla^2 u = 0$
- Heat equation: $\dot{u} = \nabla^2 u$
- Wave equation: $\ddot{u} = \nabla^2 u$

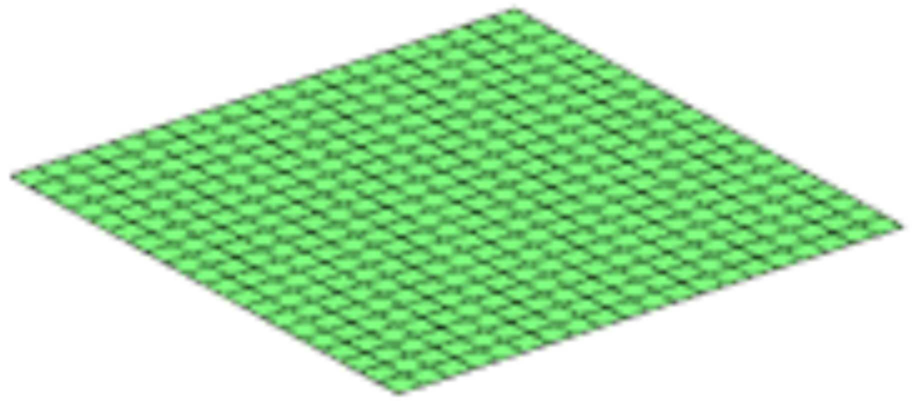
where $\nabla^2 u$ is the **Laplacian** $\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} \left[+ \dfrac{\partial^2 u}{\partial z^2} \right]$

≈ "is the average value in your neighbourhood larger than your value"





https://commons.wikimedia.org/wiki/File:Heat.gif

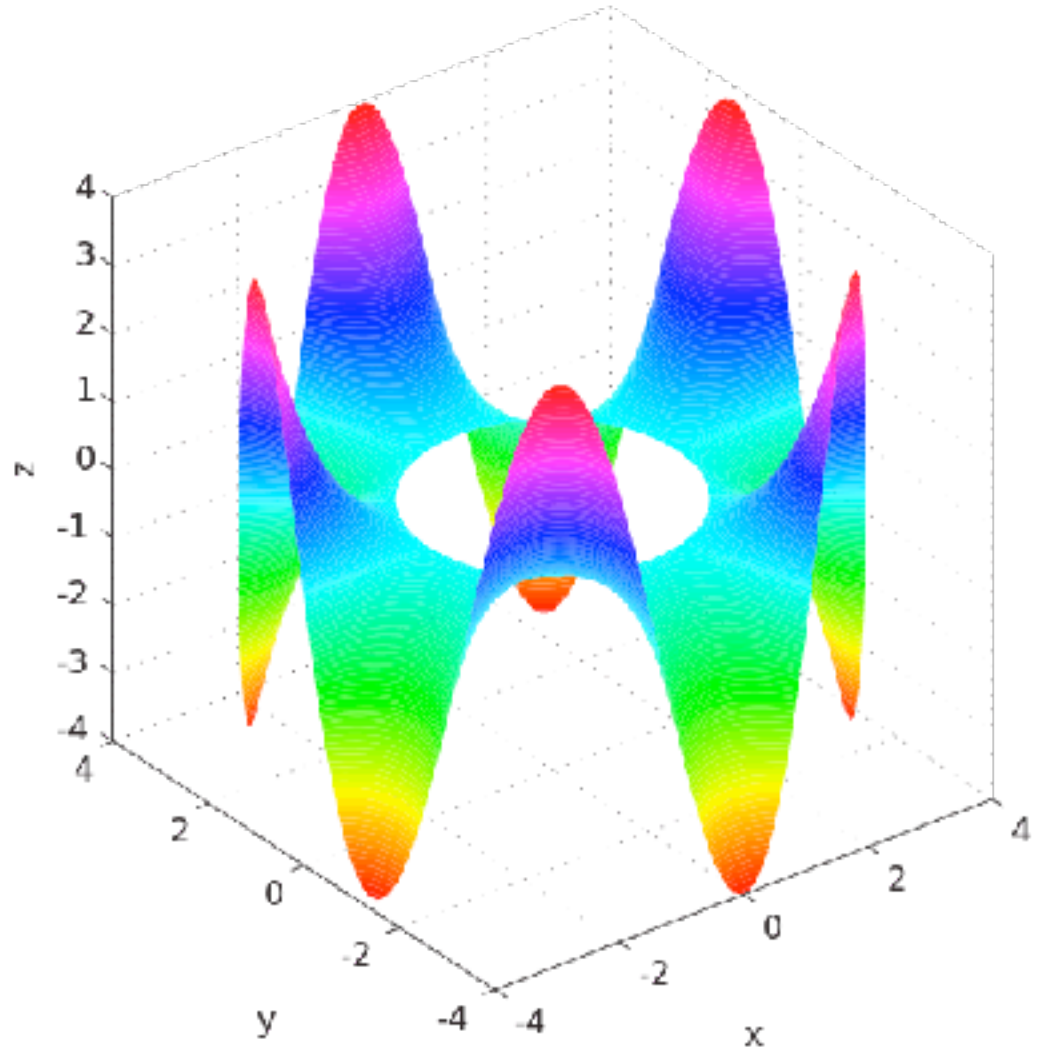


https://commons.wikimedia.org/wiki/File:2D_Wave_Function_resize.gif

Laplace equation: $\nabla^2 u = 0$

"Every point should be equal to the average in its neighbourhood"

$\Rightarrow$ *u* should be as smooth as possible given the boundary conditions



Diffusion curves

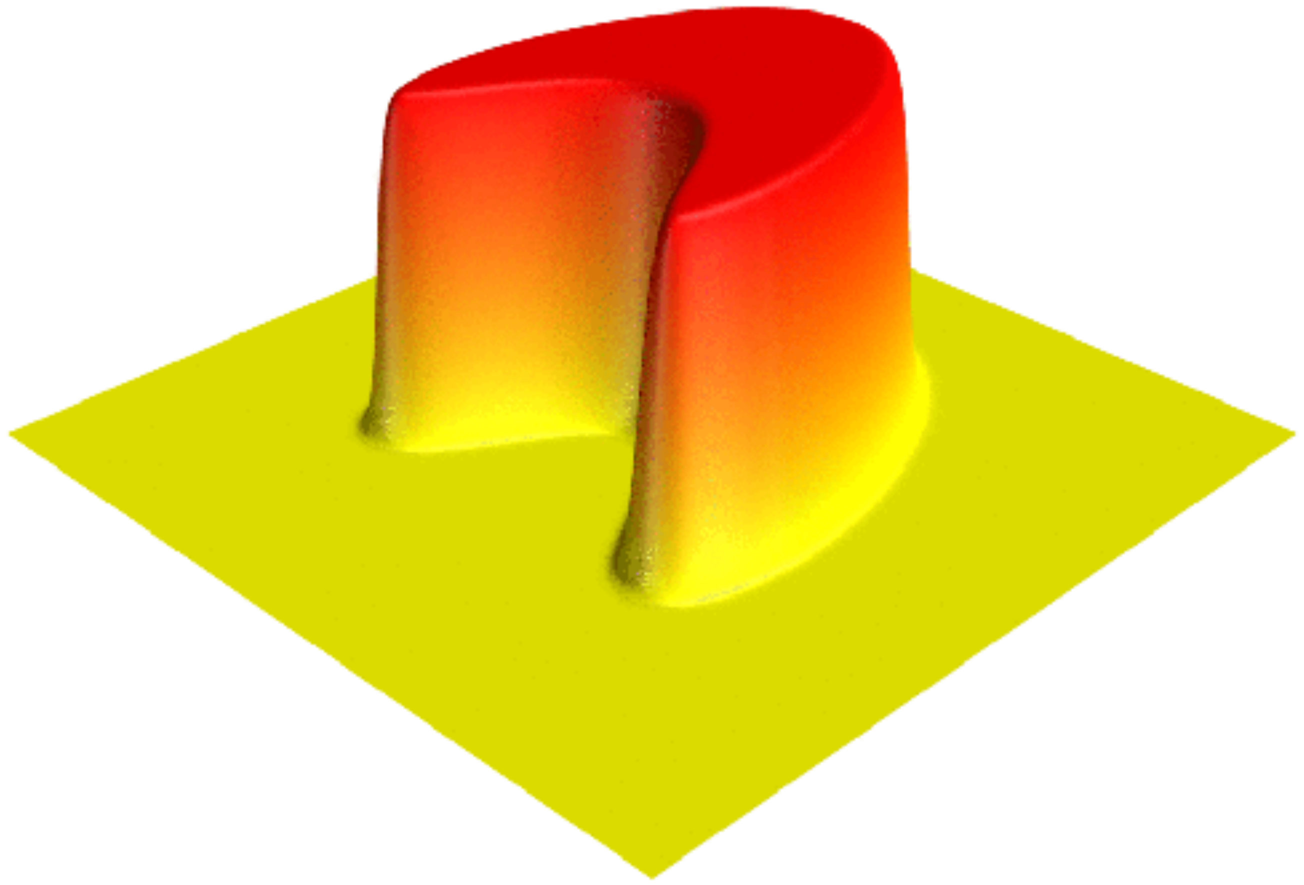
Pérez et al. 2003

Poisson image editing

https://vimeo.com/245424174

Smith et al. 2018

Heat equation: $\dot{u} = \nabla^2 u$

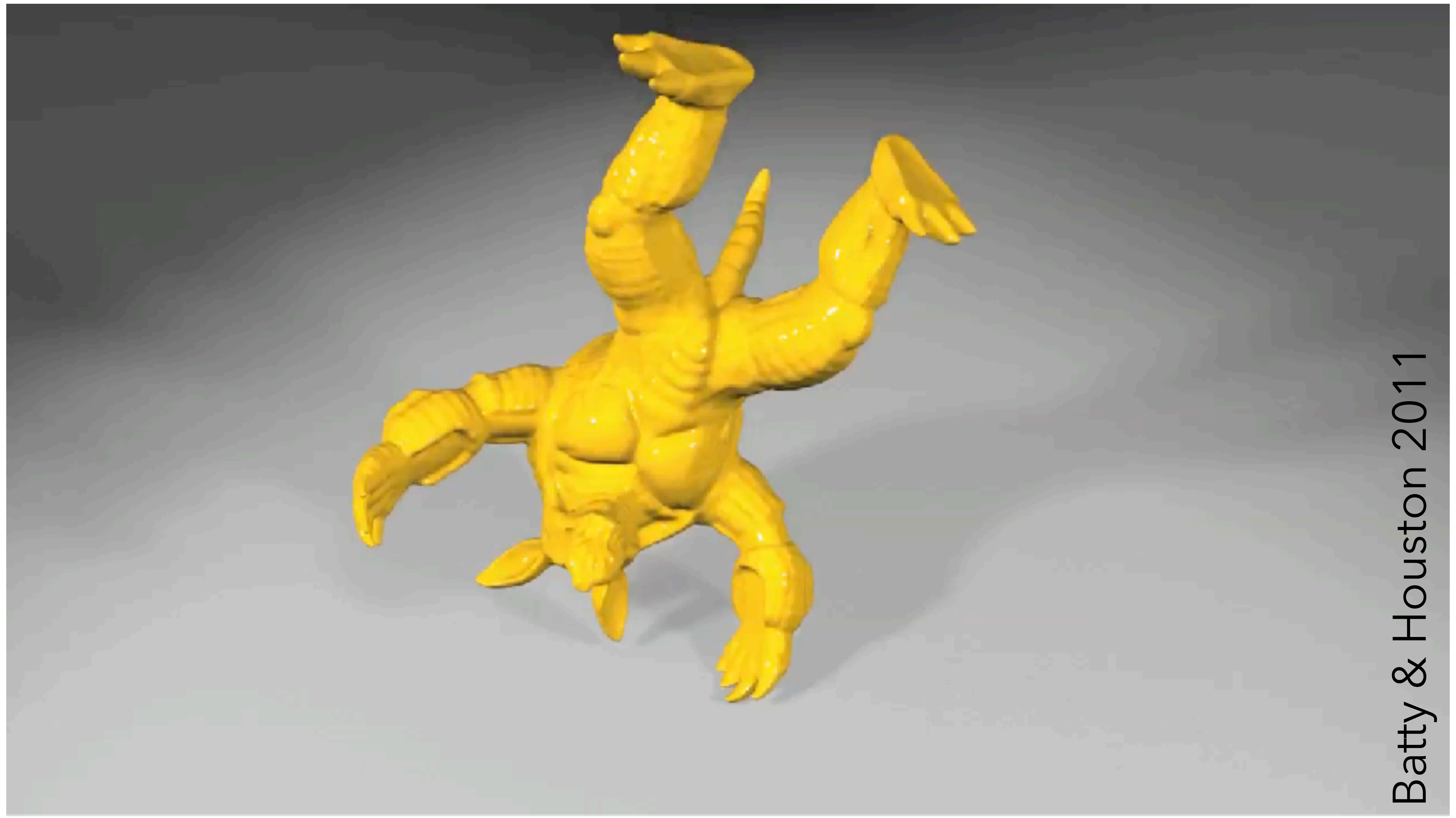


"Every point should drift towards the average in its neighbourhood"

Describes quantities getting smoothed out over time, e.g.

- Temperature
- Velocities
- Smoke density

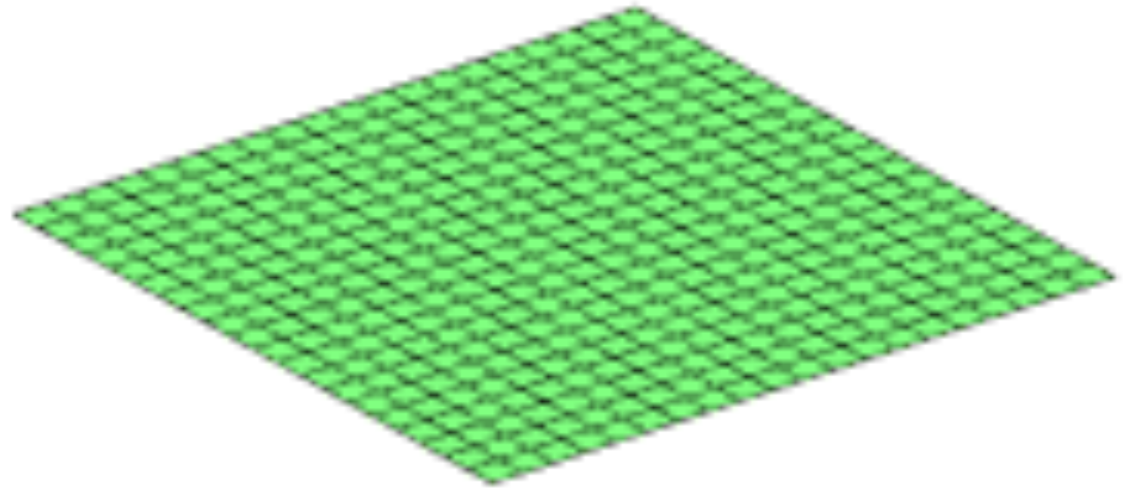
Approaches solution of Laplace equation as $t \to \infty$



Batty & Houston 2011

https://www.youtube.com/watch?v=f4H7dzFxmVw

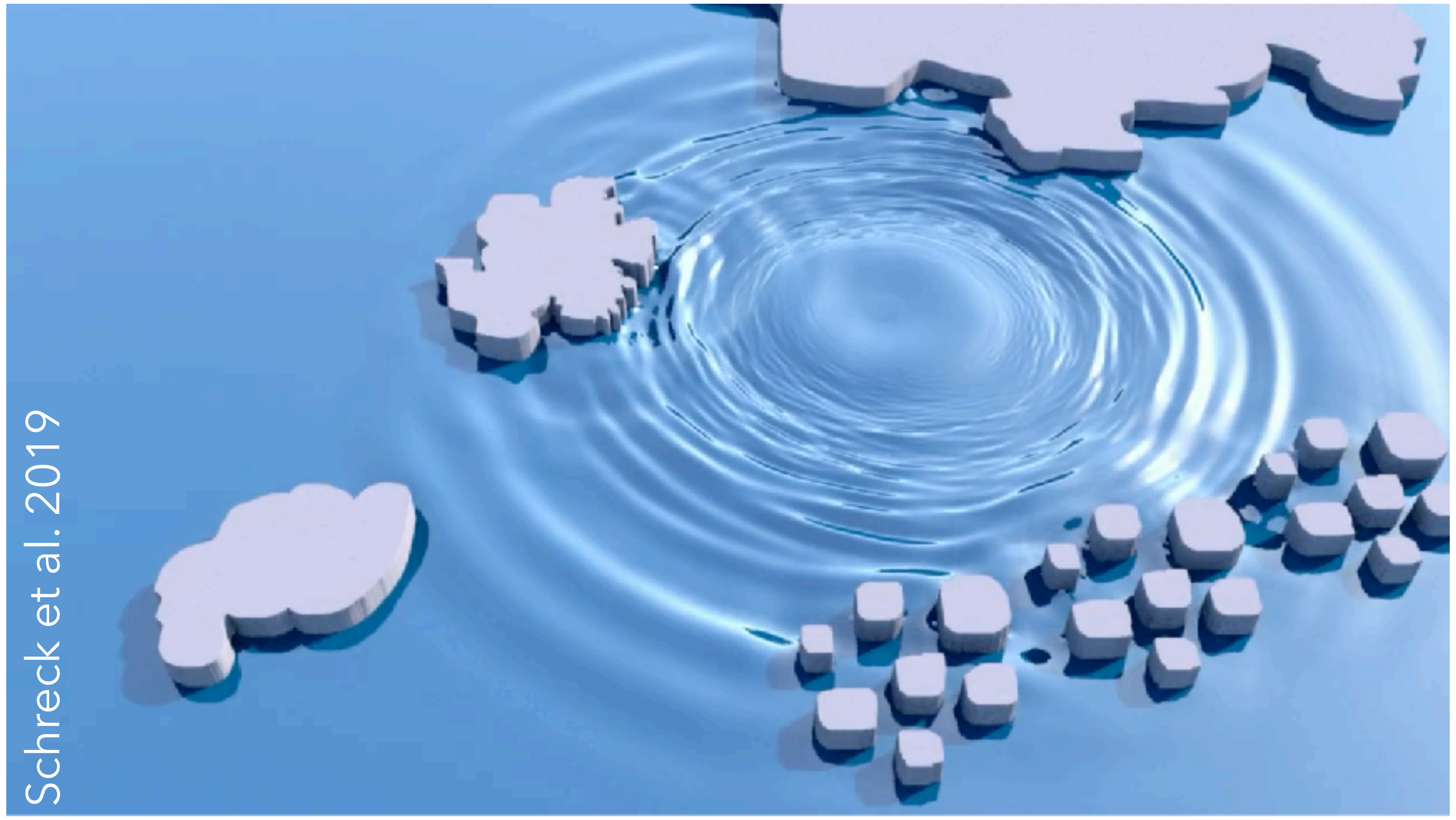
Wave equation: $\ddot{u} = \nabla^2 u$



"Every point experiences a force pulling it towards the average in its neighbourhood"

Restoring force ⇒ oscillations!

- Water waves, but also
- Elastic solids
- Cloth, hair, etc.



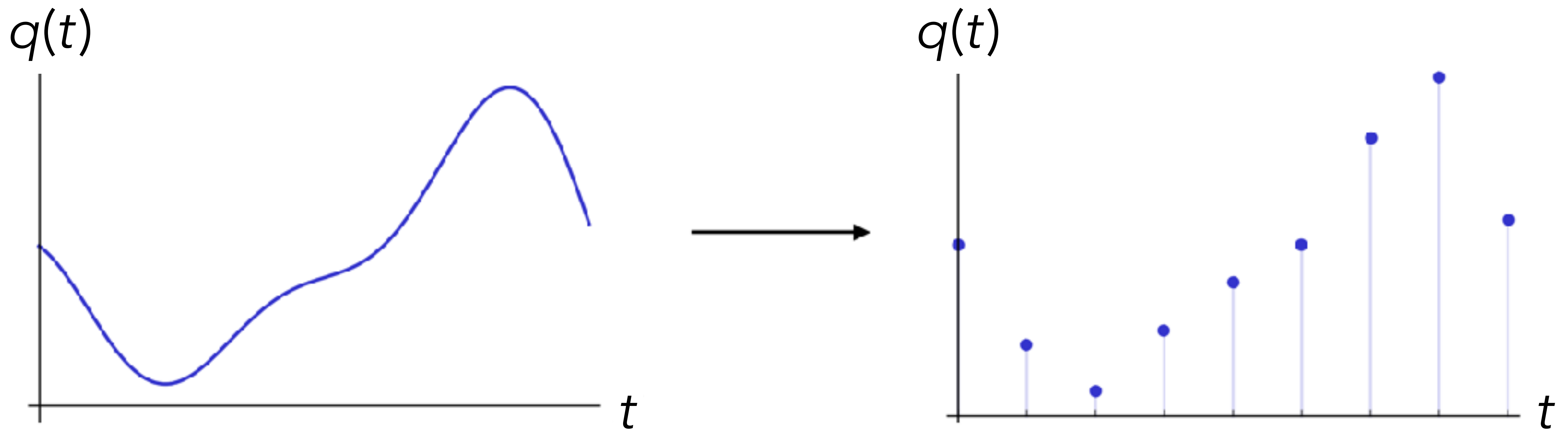
Schreck et al. 2019

https://www.youtube.com/watch?v=2ZhRNoIbf0g

So, **how do we solve a PDE?** Let's see how we solved ODEs before:

$$\frac{\mathrm{d}q(t)}{dt} = f(t, q(t))$$

Discretize! $q(t)$ is a smooth function of time → q is known at discrete instants $t_0, t_1, t_2, \ldots$

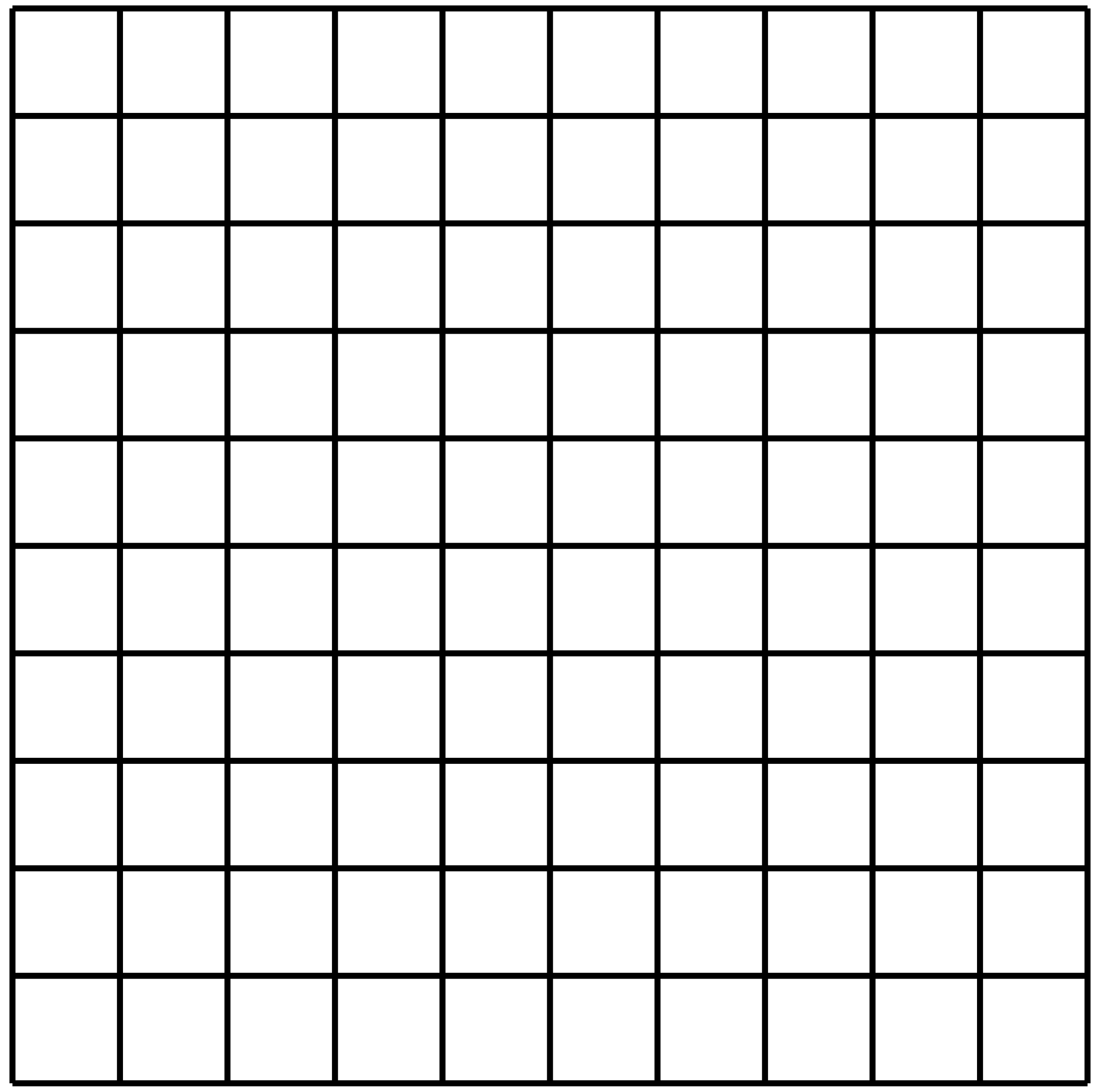


Approximate derivatives from known samples, e.g.:

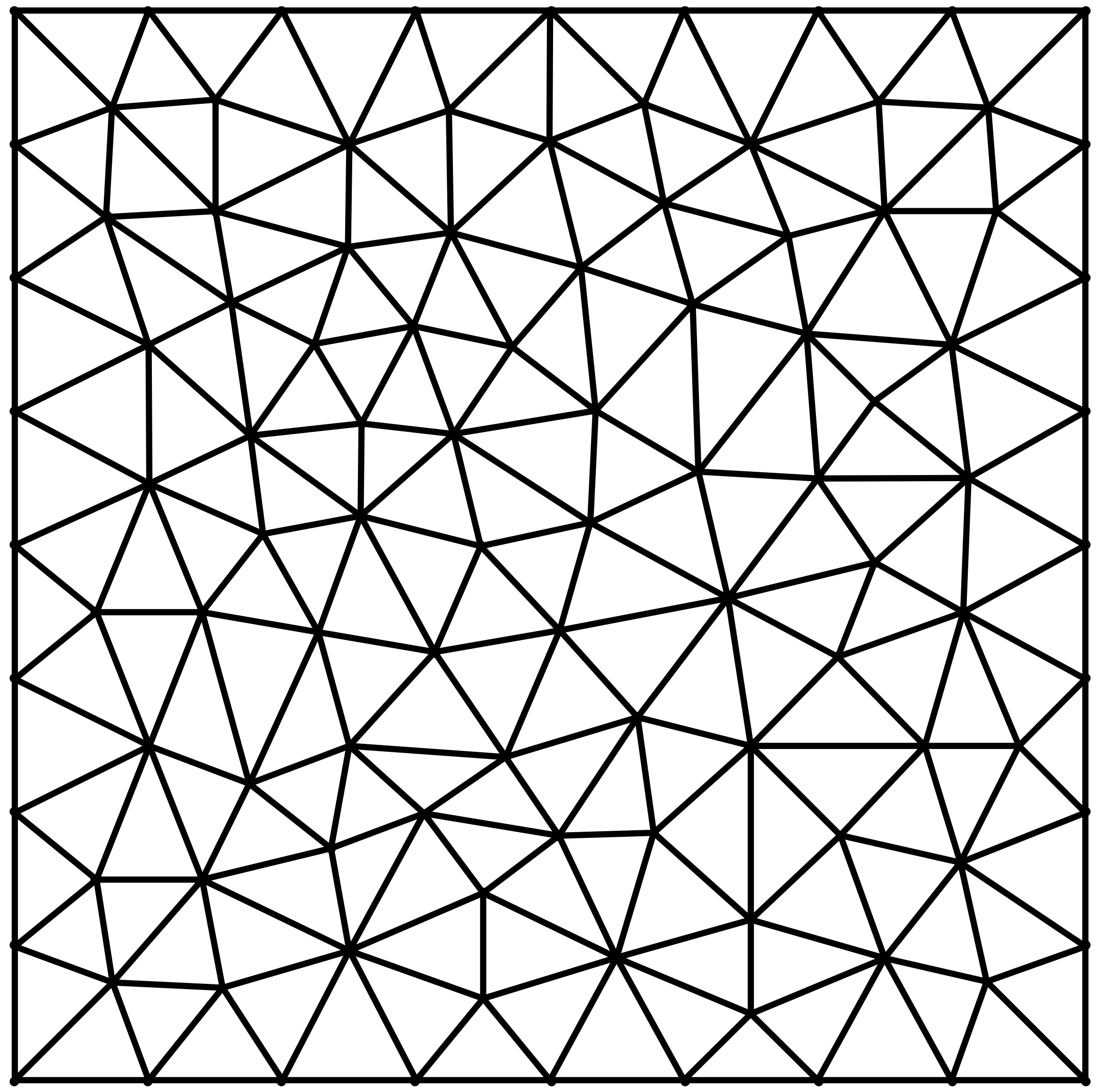
$$\frac{\mathrm{d}q}{\mathrm{d}t}(t_n) \approx \frac{q_{n+1} - q_n}{\Delta t}$$

Spatial discretizations

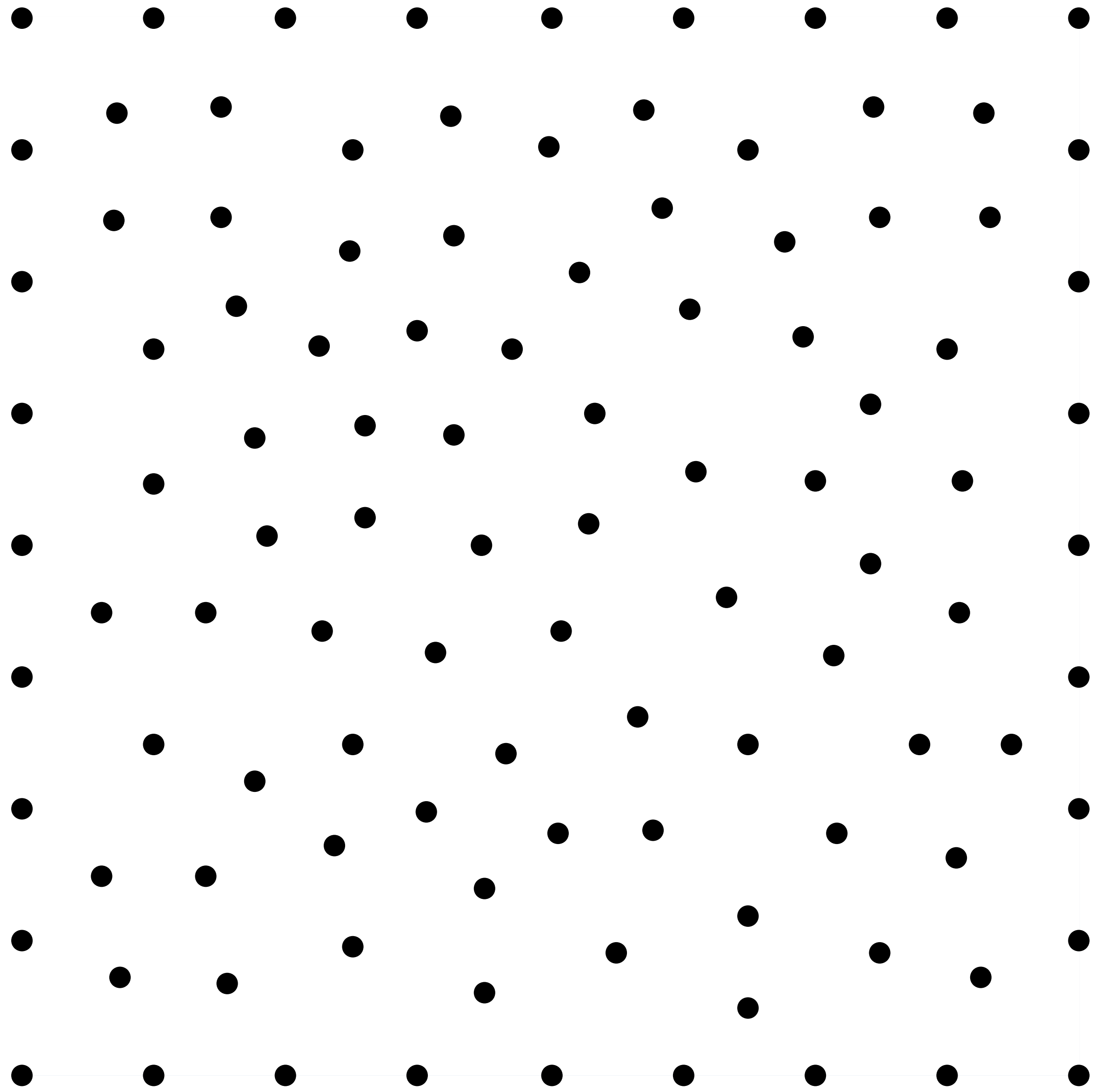
If $u(t, x, y)$ is a function of both time and space, we need to discretize space as well.



Grids



Meshes



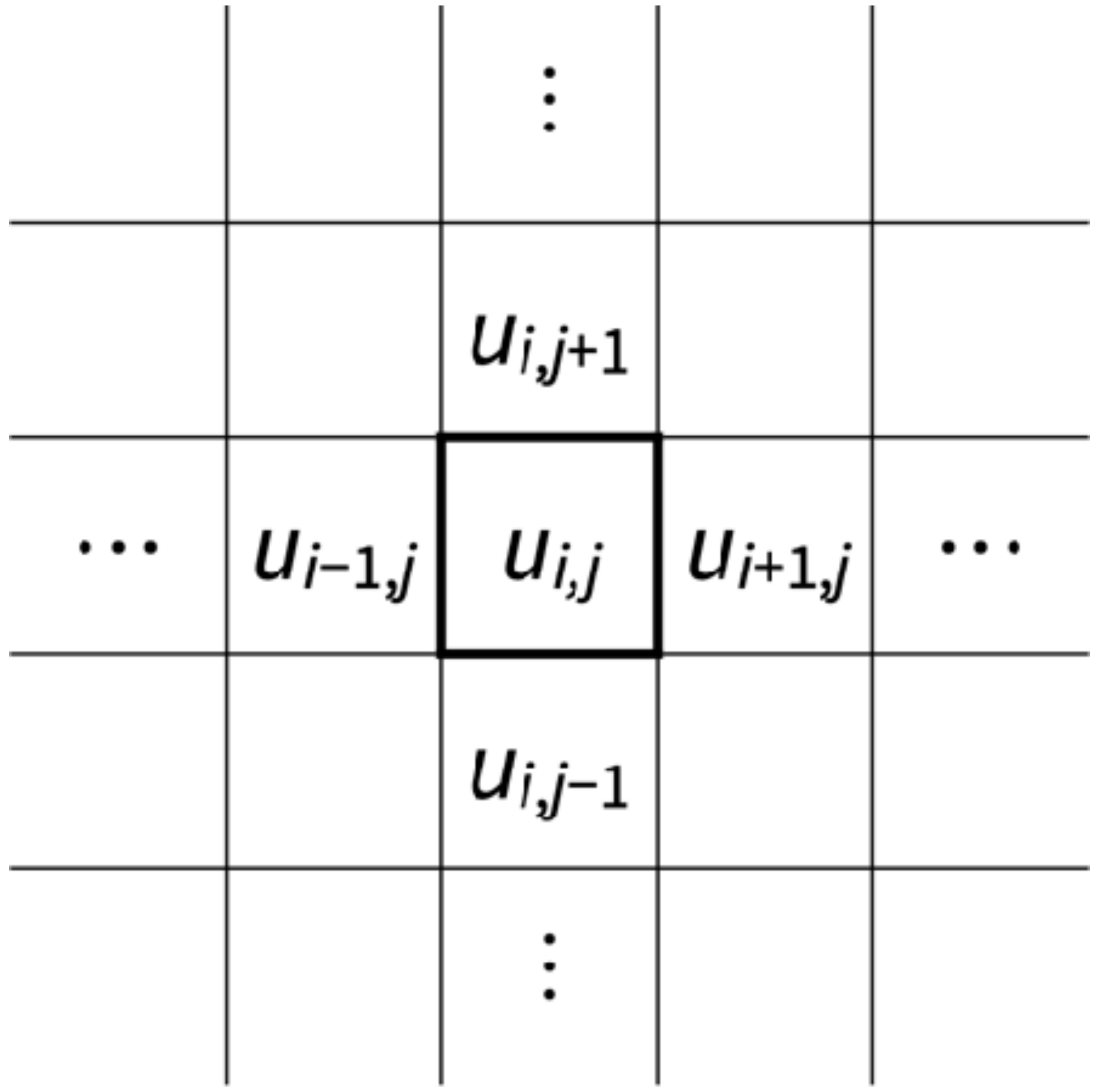
Particles

Easier computation ⟷ More flexibility

Approximating derivatives on grids is easy: **finite differences**

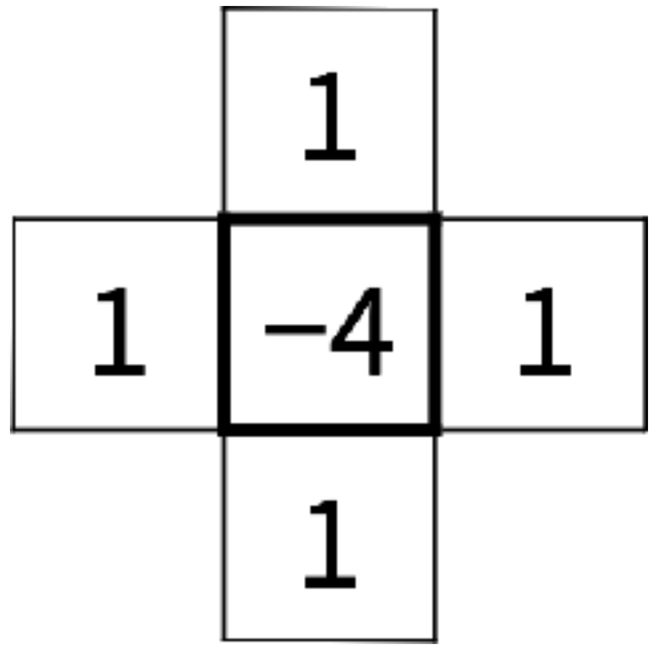
$$\frac{\partial u}{\partial x} \approx \frac{u_{i+1,j} - u_{i,j}}{\Delta x} \text{ or } \frac{u_{i,j} - u_{i-1,j}}{\Delta x} \text{ or even } \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x}$$

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right) \approx \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{\Delta x^2}$$



So the Laplacian becomes

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \approx \frac{u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j}}{\Delta x^2}$$



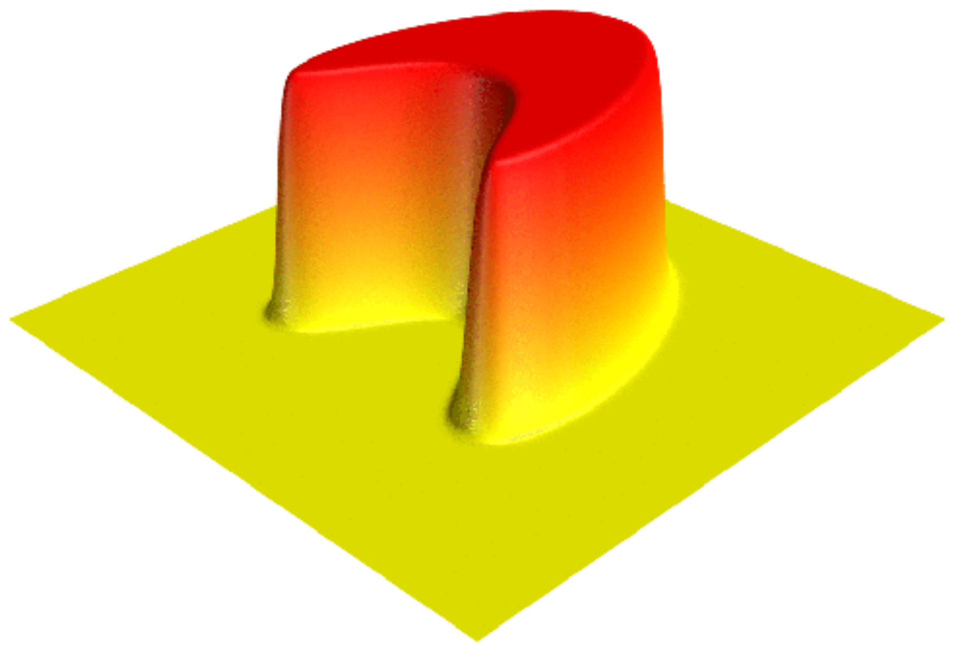
Let's apply this to the heat equation, $\dot{u} = \nabla^2 u$.

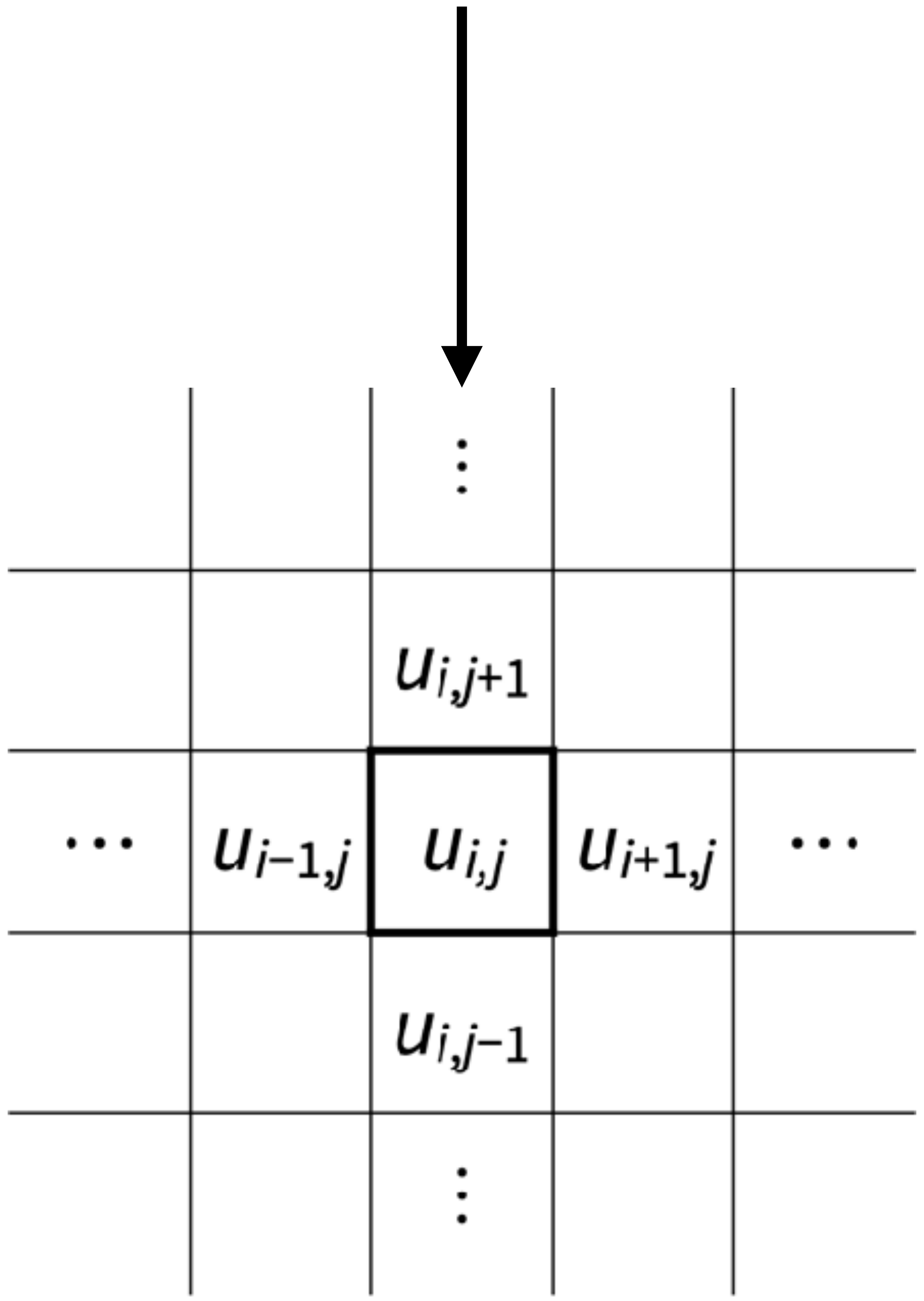
$$\dot{u}_{i,j} = \frac{u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j}}{\Delta x^2}$$

Now this is just an n^2-dimensional ODE!

Apply your favourite time integration method. Just be careful about stability…

Question: At grid edges ($i = 0$, etc.), the Laplacian formula will try to access out-of-range values. What should you do?





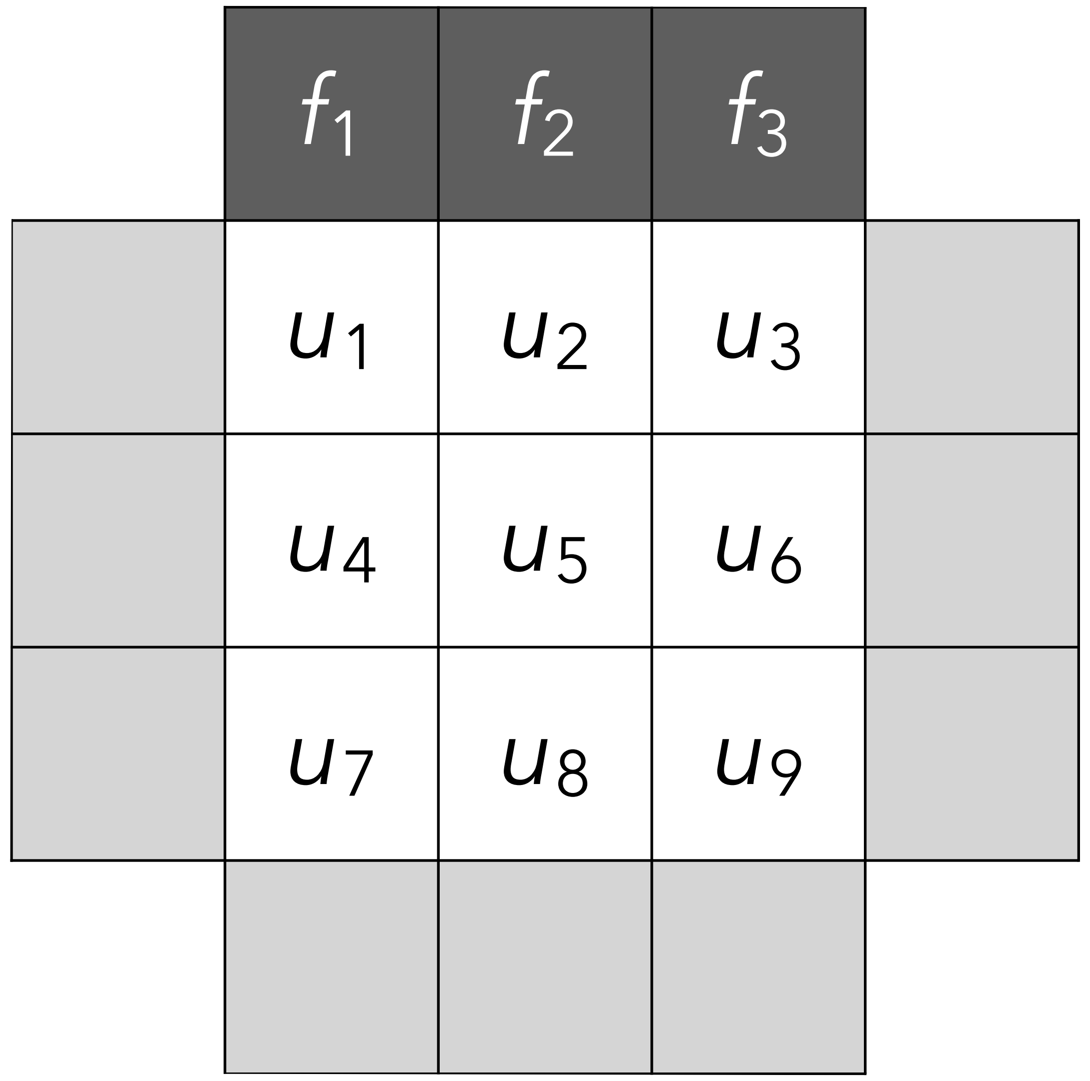
How about implicit integration?

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} = \frac{u_{i-1,j}^{n+1} + u_{i+1,j}^{n+1} + u_{i,j-1}^{n+1} + u_{i,j+1}^{n+1} - 4u_{i,j}^{n+1}}{\Delta x^2}$$

$$\left(4+\frac{\Delta x^2}{\Delta t}\right)u_{i,j}^{n+1} - u_{i-1,j}^{n+1} - u_{i+1,j}^{n+1} - u_{i,j-1}^{n+1} - u_{i,j+1}^{n+1} = \frac{\Delta x^2}{\Delta t}u_{i,j}^{n}$$

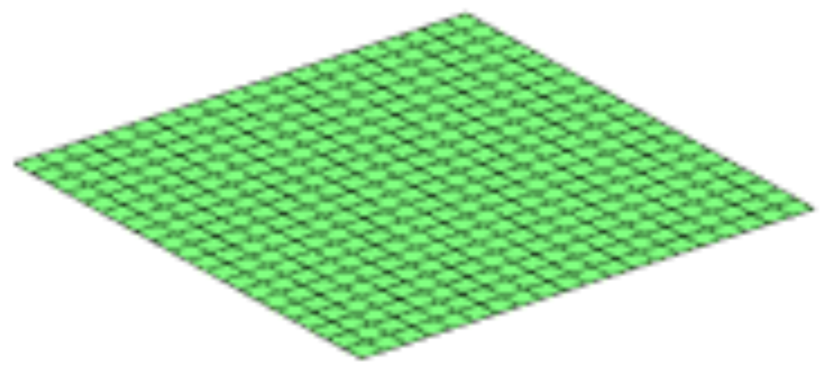
$$\begin{bmatrix}
3+\frac{\Delta x^2}{\Delta t} & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 4+\frac{\Delta x^2}{\Delta t} & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & -1 & 3+\frac{\Delta x^2}{\Delta t} & 0 & 0 & -1 & 0 & 0 & 0 \\
-1 & 0 & 0 & 3+\frac{\Delta x^2}{\Delta t} & -1 & 0 & -1 & 0 & 0 \\
0 & -1 & 0 & -1 & 4+\frac{\Delta x^2}{\Delta t} & -1 & 0 & -1 & 0 \\
0 & 0 & -1 & 0 & -1 & 3+\frac{\Delta x^2}{\Delta t} & 0 & 0 & -1 \\
0 & 0 & 0 & -1 & 0 & 0 & 2+\frac{\Delta x^2}{\Delta t} & -1 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & -1 & 3+\frac{\Delta x^2}{\Delta t} & -1 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 2+\frac{\Delta x^2}{\Delta t}
\end{bmatrix}
\begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ u_4^{n+1} \\ u_5^{n+1} \\ u_6^{n+1} \\ u_7^{n+1} \\ u_8^{n+1} \\ u_9^{n+1} \end{bmatrix}
=
\begin{bmatrix} \frac{\Delta x^2}{\Delta t}u_1^n + f_1 \\ \frac{\Delta x^2}{\Delta t}u_2^n + f_2 \\ \frac{\Delta x^2}{\Delta t}u_3^n + f_3 \\ \frac{\Delta x^2}{\Delta t}u_4^n \\ \frac{\Delta x^2}{\Delta t}u_5^n \\ \frac{\Delta x^2}{\Delta t}u_6^n \\ \frac{\Delta x^2}{\Delta t}u_7^n \\ \frac{\Delta x^2}{\Delta t}u_8^n \\ \frac{\Delta x^2}{\Delta t}u_9^n \end{bmatrix}$$

sparse!



Wave equation $\ddot{u} = \nabla^2 u$: exactly the same thing.

Just store both u and $v = \dot{u}$ on the grid



$$\dot{u}_{i,j} = v_{i,j}$$

$$\dot{v}_{i,j} = \frac{u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j}}{\Delta x^2}$$

Summary

So you want to simulate a physical phenomenon.

1. Pick a PDE that describes it (look up a mechanics textbook)
2. Pick a spatial discretization (turns it into an ODE)
3. Pick a temporal discretization (i.e. time stepping scheme)
4. Solve!

Is it really that easy?

Yes and no. Lots of (very) subtle issues in stability and convergence!